Amendments to the Claims

- At the time of the action: Claims 1, 4, 6-15, 17-20, 23-26, 29-38, 40, and 41.

- Amended Claims: Claims 1, 12, 20, 26, and 33.

- After this response: Claims 1, 4, 6-15, 17-20, 23-26, 29-38, 40, and 41.

This listing of claims will replace all prior versions and listings, of claims in the application.

     1.   (Currently Amended) A method comprising:

examining a plurality of nodes within a media timeline, wherein:

    the media timeline is for exposure over an application programming interface (API);

    one or more said nodes reference respective media;

dividing the media timeline into one or more presentations, wherein each said presentation describes a rendering of the media for a particular interval of time, and wherein each said presentation describes a collection of software components that, when executed, provides the described rendering of the media for the particular interval of time, wherein the collection of software components include a transform and comprise at least one of a timeline source, a media source, a media session, a media engine, a source resolver, ~~and~~ or a media sink;

    ~~and~~

    ~~the method further comprises:~~

configuring at least one said node for communication of events to another said node such that a change may be made to the media timeline while the media timeline is rendered;

loading each said software component described by a first said collection;

executing each said software component described by the first said collection; and

loading each said software component described by a second said collection.

2.   (Previously Canceled).

3.   (Previously Canceled).

4.   (Original) A method as described in claim 1, wherein:

each said presentation describes a respective partial topology of software components; and

the respective partial topology is for resolving into a full topology that references each software component utilized to render a respective said presentation.

5.   (Previously Canceled).

6.   (Previously Presented) A method as described in claim 1, wherein each said software component that is described by the second said collection is loaded during the executing of the first said collection.

7.   (Original) A method as described in claim 1, further comprising receiving a request from the application over the API to render the media timeline.

8.　　(Original)　A method as described in claim 1, wherein at least one said node is configured to reference an effect to be applied to an output of said media referenced by the node.

9.　　(Original)　A method as described in claim 1, wherein at least one said node is specified as read-only.

10.　　(Previously Canceled).

11.　　(Original)　One or more computer readable media comprising computer executable instruction that, when executed on a computer, direct the computer to perform the method of claim 1.

12.　　(Currently Amended)　A method comprising:

receiving a call from an application over an API for rendering a media timeline, wherein:

the media timeline includes a plurality of nodes, wherein the plurality of nodes comprises at least a parent node and a child node;

two or more said nodes reference respective media;

the media timeline defines one or more presentations including said media;

rendering the media timeline to output each said presentation, wherein the rendering further comprises dividing the media timeline into the one or more presentations such that each said presentation describes a collection of software components utilized to render said media for a particular interval of time, wherein the collection of software components include a transform and comprise at least one of a timeline source, a media source, a media session, a media engine, a source resolver, and a media sink; and

~~the method further comprises:~~

configuring at least one said node for communication of events to another said node such that a change may be made to the media timeline while the media timeline is rendered by performing at least one of the following:

changing to a property of the at least one said node;

adding one or more additional said nodes as a child to the at least one said node;

removing one or more said nodes that are children of the at least one said node;

adding an effect to the at least one said node; and

removing an effect from the at least one said node; ~~and~~

loading each said software component described by a first said collection;

executing each said software component described by the first said collection; and

loading each said software component described by a second said collection.

13. (Previously Presented) A method as described in claim 12, wherein the rendering further comprises:

examining the media timeline.

14. (Previously Presented) A method as described in claim 12, wherein:

each said collection does not change for the particular interval of time described by a respective said presentation.

15. (Previously Presented) A method as described in claim 12, wherein:

each said presentation describes a respective partial topology of software components; and

the respective partial topology is for resolving into a full topology that references each software component utilized to provide a respective said presentation.

16. (Previously Canceled).

17. (Original) A method as described in claim 12, wherein at least one said node is configured to reference an effect to be applied to an output of said media referenced by the node.

18. (Previously Canceled).

19. (Original) One or more computer readable media comprising computer executable instruction that, when executed on a computer, configure the computer to perform the method of claim 12.

20. (Currently Amended) One or more computer-readable media comprising computer executable instructions that, when executed on a computer, direct the computer to divide a media timeline into one or more presentations, wherein:

the media timeline is for exposure via an API to one or more applications;

the media timeline includes a plurality of nodes, wherein the plurality of nodes comprises at least one of a parent node and a child node;

at least two said nodes reference respective media; and

each said presentation describes rendering of respective said media for a particular interval of time, wherein each said presentation describes a collection of software components that, when executed, provide the described rendering of said media for the particular interval of time, and wherein the collection of software components include a transform and comprise at least one of a timeline source, a media source, a media session, a media engine, a source resolver, and a media sink; and

the one or more computer-readable media further comprises computer executable instructions that, when executed on the computer, direct the computer to:

load each said software component described by a first said collection;

execute each said software component described by the first said collection; and

load each said software component described by a second said collection.

21.   (Previously Canceled).

22.   (Previously Canceled).

23.   (Original)   The one or more computer-readable media as described in claim 20, wherein:

each said presentation describes a respective partial topology of software components; and

the respective partial topology is for resolving into a full topology that references each software component utilized to provide a respective said presentation.

24.   (Original)   The one or more computer-readable media as described in claim 20, wherein at least one said node is configured to reference an effect to be applied to an output of media referenced by the node.

25.   (Original)   The one or more computer-readable media as described in claim 20, wherein at least one said node is configured for communication of events to another said node such that a change may be made to the media timeline while the media timeline is rendered.

26.   (Currently Amended) A system comprising:

a plurality of media;

a plurality of applications; and

an infrastructure layer that:

provides an API for the plurality of applications which exposes a media timeline that describes one or more presentations of the plurality of media; and

manages rendering of the one or more presentations, wherein each said presentation describes rendering of said media for a particular interval of time, and wherein each said presentation describes a collection of software components configured for dynamic loading such that the collection of software components provide the described rendering of the media for the particular interval of time, wherein the collection of software components include a transform and comprise at least one of a timeline source, a media source, a media session, a media engine, a source resolver, and a media sink, wherein the collection of software components [[and]] are loaded only when needed.

27.   (Previously Canceled).

28.   (Previously Canceled).

29.   (Previously Presented) A system as described in claim 26, wherein the collection of software components does not change for the particular interval of time described.

30.   (Previously Presented) A system as described in claim 26, wherein each said presentation:

describes a respective partial topology that reference one or more software components that, when executed, provide the described rendering; and

the respective partial topology is for resolving into a full topology that references each software component that provide the rendering.

31.   (Original) A system as described in claim 26, wherein:

the media timeline includes a plurality of nodes;

at least two said nodes reference respective said media; and

at least one said node is configured to reference an effect to be applied to an output of media referenced by the node.

32.   (Original) A system as described in claim 26, wherein:

the media timeline includes a plurality of nodes;

at least two said nodes reference respective said media; and

at least one said node is configured for communication of events to another said node such that a change may be made to the media timeline while the media timeline is rendered.

33.   (Currently Amended) A timeline source comprising computer instructions that, when executed by a computer, provide:

means for dividing a media timeline into one or more presentations each describing a rendering of one or more media during a particular interval of time, wherein the media timeline exposes a plurality of nodes to [[the]] a plurality of applications, wherein one or more said nodes reference respective said media, and wherein the media timeline is configured for dynamic loading such that metadata included in at least one said node specifies a collection of said nodes to be loaded when the media timeline is rendered;

means for determining a topology for each said presentation, wherein the topology references a collection of software components that, when executed, provides the rendering; and

media processor means for executing the topology for each said presentation that is described by the media timeline.

34.   (Original) The timeline source as described in claim 33, wherein each said collection does not change for the particular interval of time described by a respective said presentation.

35.   (Original) The timeline source as described in claim 33, wherein the topology is a partial topology is for resolving into a full topology that references each software component utilized to provide a respective said presentation.

36. (Original) The timeline source as described in claim 33, wherein at least one said node is configured to reference an effect to be applied to an output of said media referenced by the node.

37. (Original) The timeline source as described in claim 33, wherein at least one said node is configured for communication of events to another said node such that a change may be made to a property of the at least one said node while the media timeline is rendered.

38. (Original) The timeline source as described in claim 33, wherein the media timeline is configured for dynamic creation such that at least one said node is created while the media timeline is rendered.

39. (Previously Canceled).

40. (Original) The timeline source as described in claim 33, wherein at least one said node is specified as read-only.

41. (Original) The timeline source as described in claim 33, further comprising means for translating a time specified by one said node for rendering the one said node with respect to a time specified by another said node.